HICSS-34 Minitrack on Collaborative Problem Solving Environments, Maui, Hawaii, January 3-6, 2001

Supporting Scientific Analysis within Collaborative Problem Solving Environments

Velvin R. Watson
NASA Ames Research Center
vwatson@ arc.nasa.gov

Abstract

This paper first lists some features of scientific analysis tools that are important for effective analysis in CPSEs (collaborative problem solving environments). Next, design criteria for achieving these features are presented. Then requirements for a CPSE architecture to support these design criteria are listed. Some proposed architectures for CPSEs are reviewed and their capabilities to support these design criteria are discussed. The most popular architecture for remote application sharing, the ITU (International Telecommunication Union) T.120 architecture, does not support highly interactive, dynamic, high resolution graphics.

A popular scientific analysis tool that conforms to the design criteria has been integrated into a collaborative environment and tested for effectiveness. The tests showed that the tool was highly effective for both synchronous and asynchronous collaborative analyses

1. Introduction

Considerable research [1] – [28] is being conducted on collaborative problem solving environments (CPSEs) because of the major benefits expected. In the past, major attention has been given to video, audio, whiteboards, chat rooms, and document sharing. However, for a scientist, the tools that he/she commonly uses for analysis are often more important than any of these. Therefore, CPSEs for scientists should also incorporate the analysis tools commonly used by the scientists.

Today, scientists are beginning to appreciate the benefits of highly interactive, high resolution, dynamic, 3D graphical representations of their data or simulations. These representations permit a better understanding of the data or simulations and often yield information that would not have been discovered with simpler displays. For many scientific applications, high performance graphics is not only appreciated, it is critical for effective understanding of complex data. Fortunately, high performance graphics can now be included in scientific analysis tools because of the major improvements in computer graphics, even on the personal computers. Therefore, future scientific analysis tools should take advantage of the benefits of high performance graphics now available.

Some very effective tools utilizing high performance graphics have been developed for scientific analysis in a single user mode. Examples are Tecplot^{TM}, IDL^{TM}, Ensight^{TM}, and FieldView^{TM}. Unfortunately, most of the tools currently used for scientific analysis cannot be easily modified to work well within CPSEs. In addition, some proposed CPSE architectures will not support high performance computer graphics in a collaborative mode. In the future, to take advantage of the major benefits of collaboration using CPSEs, it will be important to design scientific analysis tools so they can be used effectively within CPSEs and to design CPSE architectures to support the desired features of scientific analysis tools.

This paper lists the features desirable in a collaborative analysis tool. Then the design criteria for creating analysis tools that can provide these features are specified. Next, requirements of CPSE architectures to support these design criteria are discussed. Some proposed architectures for collaborative CPSEs are reviewed and their capabilities to support the specified design criteria are discussed. Finally, an example to illustrate that the specified design criteria will provide the desired features in a collaborative scientific analysis tool is presented. The example consists of the integration into a CPSE of a commonly used scientific analysis tool that was created with these design criteria. This system has been tested between sites in the U.S. and between sites on different continents. The results of these tests are presented and the performance is correlated with the design criteria.

2. Features desired in collaborative scientific analysis tools

CPSEs for scientists should provide for both synchronous and asynchronous collaborative analysis. Desirable features for effective synchronous collaboration during scientific analysis are:

1. A user-computer interface with highly interactive, high resolution, dynamic, 3D graphics. Although high performance graphics is not required by all analysis tools, most could be significantly enhanced with the use of high performance graphics.
2. All scientists should be able to see the same view of the analysis simultaneously. Having individually controllable views is also desirable, but an ability to synchronize views is very important.
3. Control of the analysis should be transferable between scientists.
4. The system should have nearly the same system responsiveness as if the tools were being operated in the stand-alone mode.
5. The system should provide the same quality of views of the analysis as if the tools were being operated in the stand-alone mode.
6. The scientists should be able to conduct the collaborative session using a network bandwidth commonly available to colleagues.

Additional features desirable for asynchronous collaboration during scientific analysis are:

1. Scientists should be able to record segments of an analysis session or all of an analysis session and post these for others to replay either collaboratively or individually.
2. Scientists should be able to easily edit and concatenate these session recordings.
3. Remote scientists should be able to easily replay these analysis sessions.
4. Remote scientists should, during replay of these analysis sessions, be able to modify or extend the analyses with their own “what if” analyses and post these sessions for others to replay.

Most scientific analysis tools available today cannot be easily modified to provide these features.

3. Design criteria to achieve the desired features

Some design criteria for providing the above features are:

1. Utilize the high performance graphics becoming available to provide highly interactive interfaces and to represent the data or simulations with high resolution, dynamic, 3D scenes.
2. Provide a capability for recording a journal file of any segment of an analysis session.
3. Provide a capability for controlling the analysis tool with a journal file. (i.e., a capability to replay an analysis session or any segment of a session.)
4. Provide a capability to easily edit and concatenate the journal files.
5. Provide a capability to condense journal files to contain only the commands needed for effective playback.

4. Requirements of CPSE architectures to support the design criteria

In order for an architecture to support the specified design criteria, it will need to provide the following:

4.1. Low latency communication between remote sites

For synchronous collaboration with highly interactive computer-user interfaces, the displays for all sites must be nearly synchronous so that the dynamical events being displayed can be discussed as they occur. For example, one scientist may want to point out a dynamical feature and count down to when it appears, “three, two, one, now”. The dynamic feature should appear at all sites close enough to the “now” to be discernable. Therefore, the events controlling the displays must be passed to all remote sites with low delay. It is unlikely that web-based architectures that pass all communications using browsers through HTTP servers (for example, with CGI) will be able to pass events between sites with sufficiently small delays to provide suitably synchronized displays. On the other hand, the tests with RemoteFAST [22] (described in sections 6.2 and 7) show that an architecture that provides dedicated event handling ports between the remote sites serviced by continuously running daemons does provide reasonably synchronized displays.

Latency between user controls and displays is even more critical. Rapidly manipulating dynamic scenes, similar to manipulating flight simulators, requires a very close coupling between the display and the controls. Therefore, for highly interactive analyses, local displays should be driven directly by local controls rather than driven via round trip communications with a remote site. Architectures requiring remote round trip communications between controls and displays are not likely to offer rapid, precise control of the display. Tests with RemoteFAST [22] (described in sections 6.2 and 7) show that displays can be controlled rapidly and precisely when the display is controlled directly by the local workstation.

4.2. Use of intelligent, compact communication between remote sites

Architectures that process high resolution scenes into pixels at one site and then send pixels to all remote sites cannot currently provide the dynamic scenes that are desirable or sometimes even required for some types of scientific analyses. There will be situations where sending pixels to remote sites may be the best approach, but the designer should be aware that selecting this approach eliminates the possibility of providing high resolution, dynamic graphics with the network bandwidths that are commonly available between scientists. The basic problem is that scientists currently receive greater than 10^9 bits per second of visual information from their workstation screens (24 color bits per pixel, 10^6 pixels per frame, and 60 frames per second) whereas the bandwidth that is commonly available between scientists is orders of magnitude less. Furthermore, scientists are reluctant to permit too much data compression for fear of creating visual artifacts that may misrepresent their data. Even desktop video conferencing which can use much greater compression has not been widely accepted because of the low quality of the video pictures over networks that are typically available to scientists.

Network bandwidths are increasing rapidly, and some argue that networks will soon provide the bandwidths to send pixels over the network fast enough to equal the information bandwidth between the workstation and the user. However, it is not likely that this will occur very soon because the number of users sharing this increased bandwidth will increase and the information bandwidth between the workstation and each user will also increase. The increasing workstation processing power will be used to provide larger and more sophisticated displays (such as autostereoscopic displays) and other types of sensory information. We are not close to exceeding the information processing bandwidth of the human visual system. The human eye has one hundred times more receptors than pixels on the current workstation displays. Therefore, even if we knew how to efficiently map pixel information to receptors, we could increase the bandwidth between the computer and the user by one hundred without exceeding the information processing bandwidth of the human visual system.

Architectures that send window drawing commands (such as the ITU T.120 architecture [29]) also do not provide adequate performance for remote applications that require dynamic, high resolution graphics. To illustrate this, the reader can try to use Microsoft NetMeeting to share any application that uses dynamic, high resolution graphics.

As illustrated in the test results (section 7), architectures that send application specific data and events for controlling applications that run at all sites (such as RemoteFAST [22]) can provide the high resolution dynamic scenes required for scientific visual analysis if the client computers at each site can render the scenes fast enough (see section 4.3.). Fortunately, even the PCs are gaining the ability to render high resolution 3D scenes rapidly.

Note that this approach for providing high performance graphics in a collaborative mode requires distribution of the data to be analyzed to the remote sites. For extremely large data sets, distributing all of the data to the remote sites may not be feasible. For this case, a typical approach is to execute the analysis and scene rendering programs on a computer closely linked with the data. Then the rendered scenes are sent to the remote sites in the form of pixels or drawing commands. Since this method does not permit high performance graphics over connections that are commonly available to scientists, this method is recommended only for locating critical subsets of the data that can be distributed to remote sites for further analysis with high performance graphics.

4.3. Sufficiently powerful computers at each remote site

The discussion of the previous requirement pointed out the need for adequate computer power at each user site to provide rapid rendering of high resolution 3D scenes. In addition, future computer-user interfaces will call for voice recognition, user tracking, user awareness, support for haptic devices, and other features that will become available in the future. These features typically require substantial computing power and flexibility --- much more than is provided by the typical “thin client” computers. For scientific research, it is still highly cost effective to invest in improving the computer-user interface. Therefore, designing scientific analysis tools with computer-user interfaces for "thin clients" or other "lowest common denominator clients" is not recommended.

4.4 Event recording and playback

The benefits achieved with event recording and playback are described in the test results (section 7). For applications that do not provide event recording and playback, it may be possible to wrap the application so that event recording and playback is done by the wrapper.

5. Proposed architectures for CPSEs

The most popular architecture for remote application sharing, ITU (International Telecommunication Union) T.120 [29], is used in NetMeeting and other commercial CPSEs. This architecture is based on running the application on one computer and sending window drawing commands to all other sites. As discussed in section 4.2, this architecture does not provide adequate support for remote collaboration requiring high resolution, dynamic scenes. There are applications that don't require high performance graphics, and the ITU T.120 architecture may be appropriate for these applications. However, it is preferable to use a CPSE architecture that supports all of the analysis tools that a scientist will need. Most scientists will find that high performance graphics is useful in at least one of their analysis tools.

Architectures such as WebFlow [10] are based on running the applications at all sites with dedicated communications for passing events between the applications. WebFlow uses Java technology and is the basis for the DOD's (Department of Defense) portal to high performance computing, the Gateway [9], and the commercial package, Tango [24]. The test of RemoteFAST [22], described in section 7, illustrates that this method does provide the high resolution dynamic scenes required for scientific visualization.

Many of the CPSE systems use Web browsers and CGI with HTTP servers for all communications. This method is unlikely to provide the low latency required for collaboration with applications that use highly interactive computer-user interfaces. (See section 4.1. for the requirement for low latency communications)

The DOE (Department of Energy) is developing many building blocks that can be used for CPSEs. For example, some of the basic communications needs for CPSEs, such as reliable multicast, are being developed in the Collaboratory Interoperability Framework Project (CIF) [2]. The Common Component Architecture Project (CCA) [1], recently developed an implementation of their CCA specification which provides components that are especially suited for scientific research.

6. Integration of a tool with the specified design criteria into a CPSE

FAST (Flow Analysis Software Toolkit) [21], is a popular tool used for visual analysis of computer simulations of physics, and it conforms to the specified design criteria. It was integrated into a CPSE as RemoteFAST (for synchronous collaborations) and as FASTexpeditions (for asynchronous collaborations) to illustrate how the design criteria listed in section 3 can provide the desired features listed in section 2. How FAST achieved the design criteria is discussed first followed by a discussion of how FAST was integrated into a CPSE by taking advantage of these design criteria.

6.1. How FAST achieved the design criteria

FAST was designed for high performance, high resolution, dynamic 3D visual analysis of computer simulations of complex physics, and it is a popular tool for analysis in computational fluid dynamics. To achieve high performance, FAST launches parallel tasks that are all controlled by a central hub.

The highly interactive, dynamic, high resolution, 3D graphical user interface is achieved by utilizing efficient event handling within the parallel tasks, and by using computers with high performance 3D scene rendering. FAST currently runs only on SGI, SUN, and HP workstations. However, even PC's can now be obtained with high performance 3D scene rendering, so this design criteria no longer requires very expensive workstations.

The capability for recording journal (script) files of analysis sessions is achieved by having each parallel task report events to the controlling central hub and having the hub record the events. The event handler in each parallel task does not directly cause actions within the task. Instead, the handler sends an ASCII text representation (command script) of the action to the hub. The hub in turn records this script in a journal file and then sends the command script back to the parallel task for execution.

The journal file playback (analysis session playback) capability is achieved by having the hub read the ASCII script from a journal file and send the command scripts to the appropriate parallel tasks. The tasks do not know whether the command scripts sent to it from the hub are the result of an event from within the task or from the hub's reading of a journal file during an analysis session replay. An advantage of this is that the scientist can play segments of a previously conducted analysis, then continue with current "what if" analyses, and follow with other segments of previously conducted analyses.

The capability to easily edit and concatenate the journal files is achieved by making the script commands ASCII text. Therefore, the files can be modified with a word processor. The script commands are documented in the FAST Users Guide [21].

The capability to condense the journal file is achieved with a special utility program. The only cause for large journal files in this system is the rapid recording of mouse movements that change the scene viewing position. These mouse movements are recorded very rapidly to provide rapid and accurate response recording. The utility program condenses the large number of very small transformations into fewer transformations that will provide equivalent smooth looking transformations on playback.

6.2. Integration of FAST into a CPSE for synchronous collaboration

To create a synchronous collaborative visualization tool, FAST was combined with a program to handle TCP/IP unicast communications between remote sites. This tool was named RemoteFAST [22]. (As soon as multicast is prevalent, it should be used instead of unicast for multipoint collaborations to eliminate the need to send multiple streams from the controlling site.) To start a synchronous session, the data to be analyzed is distributed to each site and FAST is launched at each site. Then, the program to handle communications at each site is launched to create a daemon dedicated to efficient passing of events between the sites. During the session, the controlling RemoteFAST site simply detects the script commands as they are being recorded into the journal file and sends the same script commands over the network to all controlled RemoteFAST sites. At the controlled sites, RemoteFAST simply reads the incoming script commands as though they were being read from recorded journal files on the local disk and passes them onto the RemoteFAST hub.

This technique provides many advantages. It is simple. The bandwidth between sites need not be large because only script commands are sent between sites. And, the system response experienced by the users is nearly the same as the response in stand-alone mode. The system response is very good because:

1. the dedicated communications daemons provide a nearly unnoticeable delay in sending the script commands over the network
2. intelligent, compact information (i.e., application specific data and events rather than pixels) is sent between sites
3. the 3D scene rendering is performed by the local computer.

Therefore, all remote scientists appear to be seeing the same high resolution, dynamic, 3D scenes simultaneously (see section 7 for details).

RemoteFAST is normally used along with a desktop video tool if the network bandwidth permits, or along with a normal phone conference if the network bandwidth doesn't permit the video.

These remote collaboration sessions can be recorded and posted onto the Web for other scientists to playback and modify at their convenience. See the next section for details.

6.3. Integration of FAST into a CPSE for asynchronous collaboration

To create an asynchronous collaborative visualization tool, FAST was wrapped with a C Shell script to permit use with the World Wide Web. This tool was named FASTexpeditions. The data to be analyzed and the analysis sessions (journal files) are made available from Web pages. Selecting the data from a Web page causes downloading of the data to the local computer, automatic launching of FASTexpeditions on the local computer, and execution of a script to set up the initial state of the analysis. Subsequent selections of analysis segments from the Web page causes execution of the journal files for those segments.

For most of the investigations that we have posted to the Web, all of the analysis segment journal files are packaged and downloaded along with the initial data because doing this permits playing of any analysis segment without returning to the remote Web server for the journal files. In this case, the URL used on the Web page refers to the downloaded journal files on the local disk, so the Web browser gets these immediately from the local computer disk rather than waiting for the remote Web server to respond and deliver them.

Sound files can be included in the journal files for an audio description of the analysis as it occurs.

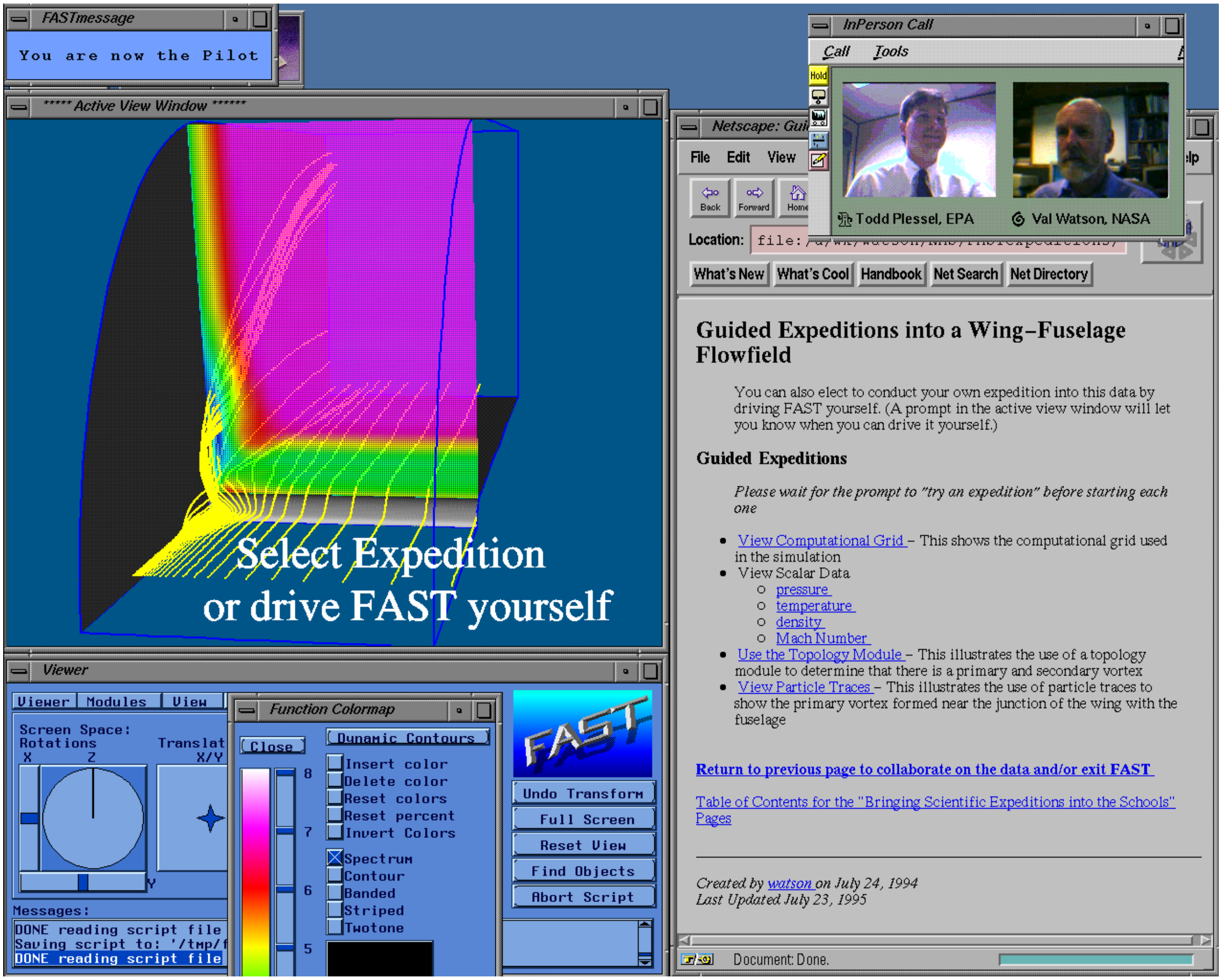
To provide safety from people who might post malicious journal files, the C Shell wrapper scans each journal file and removes unsafe commands.

To facilitate the ease of collaboratively discussing the posted analyses with remote colleagues, the Web pages containing the FASTexpeditions also contain selections for automatically initiating a synchronous collaboration using RemoteFAST (described in section 6.2).

A utility was created to automatically generate a FASTexpedition Web page with URLs pointing to the data from the computer simulation and the journal files of the analysis segments.

7. Test results

RemoteFAST and FASTexpeditions have been tested in collaborative sessions between sites within the U.S. and between sites in different continents Within the U.S., the tests were conducted primarily between the NASA Ames Research Center in California and the EPA (Environmental Protection Agency) in North Carolina. Tests between the U.S. and Australia were conducted between NASA Ames Research Center and Perth Australia. Tests between the U.S. and Europe were conducted between the EPA and Monte Carlo, Monaco or Poitiers, France. Figure 1 shows the computer screen during a session.



You are now the Pilot

Select Expedition
or drive FAST yourself

```
DONE reading script file
Saving script to: '/tmp/f
DONE reading script file
```

file: /u/wk/watson/WWB/FASTexpeditions/

Guided Expeditions into a Wing–Fuselage Flowfield

You can also elect to conduct your own expedition into this data by driving FAST yourself. (A prompt in the active view window will let you know when you can drive it yourself.)

Guided Expeditions

Please wait for the prompt to "try an expedition" before starting each one

- View Computational Grid – This shows the computational grid used in the simulation
- View Scalar Data
  - pressure
  - temperature
  - density
  - Mach Number
- Use the Topology Module – This illustrates the use of a topology module to determine that there is a primary and secondary vortex
- View Particle Traces – This illustrates the use of particle traces to show the primary vortex formed near the junction of the wing with the fuselage

Return to previous page to collaborate on the data and/or exit FAST

Table of Contents for the "Bringing Scientific Expeditions into the Schools" Pages

Created by watson on July 24, 1994
Last Updated July 23, 1995

Document: Done.

Figure 1. Illustration of a FASTexpedition and RemoteFAST session

RemoteFAST and FASTexpeditions were highly effective for both synchronous and asynchronous collaboration. The effectiveness of the collaboration was nearly as good as being together in the same office and looking at the same workstation while using FAST for the analysis or for a playback of an analysis. All of the desired features listed in section 2 were achieved.

For synchronous collaboration, the response of the visual analysis tool was nearly the same as in stand-alone mode. All sites were able to view the same high resolution (1280x1024), dynamic, 3D scenes simultaneously. Individual sites could independently control their own scene viewing position, but the viewing position could also be resynchronized with the controlling site's viewing position. Control of the analysis was easily transferred between sites. The bandwidth utilized between sites during a remote collaboration session was measured to peak at less than 1K bit/second. Note that this low bandwidth utilization and high display performance is achieved by sending script commands over the network and by having the local computer create and render the scenes. This performance cannot now be achieved by sending pixels over the network. Even systems that send scene graphs (such as VRML files) over the network do not match this performance.

For asynchronous collaboration, the analyses posted on the Web were easily downloaded and played. After the initial data download, the playback performance was identical to the performance of playback from journal files on the local disk.

Stereo glasses were often used to obtain stereoscopic scenes in both synchronous and asynchronous modes.

The major advantages of FASTexpeditions over VRML or movie files posted on the Web are:

1. The 3D display performance is superior.
2. Viewers download the actual data and can perform their own "what if" analysis on the data.
3. Viewers can modify the analyses they download and post their own analyses back on the Web.
4. Viewers can collaboratively review and modify the posted analyses with remote colleagues, and these analyses can be posted back onto the Web.

RemoteFAST and FASTexpeditions were used in conjunction with InPerson^{TM}, SGI's desktop video conference tool, whenever the network bandwidth was high enough (i.e., between France and the U.S. and between sites within the U.S.). Ordinary phones were used instead of InPerson^{TM} when the network bandwidths would not support satisfactory desktop video (i.e., between Monaco and the U.S. and between Australia and the U.S.).

The scenario used most often to demonstrate the features of FASTexpeditions and RemoteFAST follows:

1. A scientist goes to a Web site where FASTexpeditions of various analyses of computer simulations of physics are posted.
2. The scientist selects one of the FASTexpeditions and views several of the posted analyses of the data.
3. The scientist then extends the author's posted analysis with his/her own "what if" analysis.
4. The scientist then contacts the author of the posted analyses with a phone or InPerson^{TM} and asks the author about one of the features seen in an analysis.
5. The author and the scientist then both initiate a remote collaboration by making selections on the Web page to automatically start RemoteFAST.
6. The author and the scientist then use RemoteFAST collaboratively to investigate the feature.

Typically, the desktop video was only used at the beginning of the collaborative session when establishing initial contact. When the interest shifted from the initial "hello" to the analysis of the data, the primary focus was shifted to the 3D scenes of the visual analysis process and to the audio.

8. Summary

CPSEs have the potential for a major impact in scientific research. To achieve this potential, it is important to include the scientist's favorite analysis tools within his/her CPSE. However, most current scientific analysis tools cannot be easily modified to work well within a collaborative environment. Therefore, when designing future scientific analysis tools, it is important to design them to work well within a collaborative environment.

This paper presented the features desired in a collaborative analysis tool and the design criteria to achieve the desired features. The use of high performance computer graphics, now becoming available even on personal computers, was highly recommended to improve the computer interface and the representation of data and simulations within scientific analysis tools.

The features required in CPSE architectures to support the specified design criteria were presented. A review of some proposed CPSE architectures indicated that some do not support the specified design criteria. In particular, the popular ITU standard, T.120, does not support high performance graphics in a collaborative mode.

A popular analysis tool that conformed to the design criteria was incorporated in a CPSE and tested. The tests showed that the tool was very effective for both synchronous and asynchronous collaboration and that it provided all of the desired features listed.

9. Acknowledgments

FAST was created at NASA Ames Research Center by the team listed in [21]. Jean Clucas, John West, and Tim Sandstrom were especially helpful in creating FASTexpeditions and RemoteFAST[22]. Tim Sandstrom also provided valuable support for users of these programs. Todd Plessel, with the EPA in Research Triangle Park in North Carolina, installed these programs within the EPA and helped extensively with testing and with many demonstrations.

10. References

Current Research on CPSEs by Organization

Department of Energy

[1] Common Component Architecture (CCA)
(CCA provides components especially valuable for scientific research)
Contact - Rob Armstrong - Sandia
Website - http://z.ca.sandia.gov/~cca-forum/

[2] Collaboratory Interoperability Framework Project (CIF)
(CIF will provide fundamental communications needs to build on)
Contact - Deb Agarwal - LBL
Website - http://www-itg.lbl.gov/CIF/

[3] Advanced Visualization Communications Toolkit
(This will provide network aware components to optimize visualization)
Contact - Deb Agrawal - LBL
Website-http://www-itg.lbl.gov/~deba/NGI/AdvVizComm.html

[4] Corridor One Project
(High performance visualization over very high bandwidth networks)
Contact - Rick Stevens - ANL
Websitehttp://www-fp.mcs.anl.gov/fl/research/Proposals/co.htm

[5] Toolkit for Collaboratory Development
(Includes Core2000 – a collaboratory research environment based on Habanero from NCSA)
Contact – Jim Meyers – PNL
Website - http://www.emsl.pnl.gov:2080/docs/collab/

[6] DOE2000 Electronic Notebook Project
(Electronic notebook projects at PNL, LBL, and ORNL)
Main Website - http://www.epm.ornl.gov/enote/
Contacts
Jim Myers - PNL
Sonia Sachs- LBL
Al Geist – ORNL

[7] DOE2000 Collaboratory Research
(Basic collaboratory research projects)
Website - http://www-unix.mcs.anl.gov/DOE2000/collabs.html

[8] DOE2000 Collaboratory Pilot Projects
The Diesel Combustion Collaboratory
Website - http://www-collab.ca.sandia.gov/Diesel/ui/
The Materials MicroCharacterization Collaboratory
Website - http://tpm.amc.anl.gov/MMC/
Environmental Molecular Sciences Collaboratory
Website - http://www.emsl.pnl.gov:2080/docs/collab/
Fusion Collaboratory
Website - http://www.fusionscience.org/collab/REE/

Department of Defense

[9] The Gateway
(A portal to high performance computing. This uses a three tier architecture built from WebFlow and Globus)
Contact - Marlon Pierce - WPAFB
Website – http://www.osc.edu/~kenf/sciPortal

NSF and Universities

[10] WebFlow
(A visual programming paradigm for Web/Java based coarse grained distributed computing. This is based on Java technology.)
Contacts
Tomasz Haupt - Syracuse University
Wojtek Furmanski - Syracuse University
Website - http://www.npac.syr.edu/users/haupt/WebFlow/

[11] SCIRun
(This provides computational steering)
Contact - Christopher R. Johnson - University of Utah
Website - http://www.cs.utah.edu/sci/publications/scitools96/

[12] NCSA Collaboration Systems
(Habanero provides state and event synchronization for multiple copies of a software tool. It utilizes Java.)
Contact - Polly Baker - NCSA
Website - http://havefun.ncsa.uiuc.edu/

[13] PUNCH
(An Architecture for Web-Enabled Wide-Area Network-Computing)
Contact - Prof Jose Fortes - Purdue Univ
Website - http://punch.ecn.purdue.edu/

[14] DISCIPLE
(Distributed System for Collaborative Information Processing and Learning)
Contact - Dr. Ivan Marsic - Rutgers Univ
Website - http://www.caip.rutgers.edu/multimedia/groupware/

[15] Space Physics and Aeronomy Research Collaboratory
(an environment for collaborative research in space physics and aeronomy)
Contact - Gary Olson - Univ of Michigan
Website - http://intel.si.umich.edu/sparc/

[16] Stanford Interactive Workspaces
(For exploring possibilities for people to work together in technology-rich spaces)
Contact - Terry Winograd - Stanford
Website - http://graphics.stanford.edu/projects/iwork

[17] High Performance and Real Time Corba
(Research on improving the throughput perfomance and reducing latency of Corba)
Contact - Doug Schmidt - UC Irvine (Wash Univ.)
Website - http://www.cs.wustl.edu/~schmidt/corba-research-performance.html

NASA

[18] Intelligent Synthesis Environment (ISE) and Collaborative Engineering Environment (CEE)
(NASA's project to create collaborative analysis and design environments)
Contact – W Lundy, NASA Lewis Research Center, for ISE
Website - http://ise.nasa.govContact - Ed Chow, NASA Jet Propulsion Laboratory, for CEE
Website - http://ce-server.jpl.nasa.gov/

[19] Science Desk
(A project to create collaborative research environments with AI support)
Contact - Rich Keller – NASA Ames Research Center
Website - http://sciencedesk.arc.nasa.gov

[20] Mars Web Pages
(A website for the collaborative selection of Mars landing sites)
Contact - Glenn Deardorff – NASA Ames Research Center
Website – http://marsoweb.nas.nasa.gov/landingsites/

[21] FAST (Flow Analysis Software Toolkit
(A tool for visual analysis of computer simulations of complex physics)
Contact – Tim Sandstrom, NASA Ames Research Center
Website – http://www.nas.nasa.gov/Software/FAST

[22] RemoteFAST and FASTexpeditions
(Tools for asynchronous and synchronous collaborative scientific visualization)
Contact – Val Watson – NASA Ames Research Center
Website
http://www.nas.nasa.gov/Software/FAST/FASTexpeditions

Industry

[23] Intelligent Human-Computer Interaction
(An environment for collaboration based on rooms. Awareness and privacy issues are addressed.)
Contact - Samuel Bayer - Mitre Corp
Website - http://www.mitre.org/resources/centers/it/g063/hci-index.html

Commercial CPSE Systems

[24] Tango Interactive
(Based on WebFlow from Syracuse University)
Contact - Marek Podgorny - WebWisdom
Website - http://www.webwisdom.com/tangointeractive/

CPSE Organizations

[25] Computingportals
Home Page - http://www.computingportals.org/Survey of projects - http://www.computingportals.org/projects

[26] Computer Supported Cooperative Work (CSCW)
Website - http://www.acm.org/sigchi/cscw2000/index.html

Reports on CPSEs

[27] Report on Collaborative Virtual Environments 1998
University of Manchester, UK, 17-19th June 1998
Elizabeth Churchill and David Snowden
http://www.fxpal.xerox.com/ConferencesWorkshops/cve/Report.htm

[28] Workshop on CPSEs for Scientific Research
San Diego, CA., 29 June – 1 July, 1999
http://www.emsl.pnl.gov:2080/docs/cpse/workshop/index.html

CPSE Standards

[29] International Telecommunication Union's Proposed Standards
Complete listing of proposed standards -
http://www.itu.int/publications/telecom.htm
Proposed standard for application sharing –
http://www.itu.int/itudoc/itu-t/rec/t/t120.html
Proposed standard for audiovisual and multimedia systems -
http://www.itu.int/itudoc/itu-t/rec/h/h323.html